Remarks

In the Final Office Action mailed April 7, 2004:

1. Claims 1-4, 6-7, 13 and 17-19 were rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,408,360 (Chamberlain);
2. Claims 5 and 12 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Chamberlain, in view of U.S. Patent No. 6,584,548 (Bourne);
3. Claim 15 was rejected under 35 U.S.C. § 103(a) as being unpatentable over Chamberlain, in view of U.S. Patent No. 6,415,270 (Rackson);
4. Claims 8-11 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Chamberlain, in view of U.S. Patent No. 6,026,413 (Challenger);
5. Claim 14 was rejected under 35 U.S.C. § 103(a) as being unpatentable over Chamberlain, in view of U.S. Patent No. 5,802,582 (Ekanadham);
6. Claim 16 was rejected under 35 U.S.C. § 103(a) as being unpatentable over Chamberlain, in view of U.S. Patent No. 6,151,643 (Cheng);
7. Claims 20-24 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Chamberlain, in view of Ekanadham; and
8. Claim 25 was rejected under 35 U.S.C. § 103(a) as being unpatentable over Chamberlain, in view of Ekanadham and Bourne.

I. Chamberlain (U.S. Patent No. 6,408,360)

Chamberlain is directed to the automatic caching of web pages that have dynamic content (abstract; column 4, lines 21-24). Because it is directed to a different area, Chamberlain cannot anticipate or make obvious all aspects of Applicants' invention.

A. Chamberlain Requires a Cache to be Colocated with a Web Server

Embodiments of Applicants' invention (e.g., claim 1) are designed for scenarios in which a cache system or server is separate from a data server or web server that produces data to be cached in the cache system.

In contrast, in Chamberlain, a cache and a web server are colocated (Fig. 4; column 6, line 65 to column 7, line 14), in a system termed a "server caching system" (column 6, line 65). The tight coupling of Chamberlain's cached-response analyzer 306 (part of the web server) and

cache 304 (Fig. 4) allows the validity analyzer 315 (part of analyzer 306) to "automatically invalidate[ ] pages based upon whether the candidate cached response is stale" (column 13, lines 4-7), but cannot automatically notify a physically separate cache system of invalidated data.

By requiring the cache and web server to be colocated, Chamberlain *teaches away* from embodiments of the present invention.

B. Chamberlain Does Not Automatically Invalidate Cached Data in Response to a Data Request Received at the Cache

In claimed embodiments of the present invention, a data item stored in a cache system is automatically invalidated in response to a change request received at the cache system.

The Examiner stated the following (the individual citations were added):

> However, Chamberlain teaches that when server 100 receives a URL from a client, the HTTP server 206 passes the URL to the URL parser 303, which breaks the URL into different parts. The parsed URL is passed to the cache control unit 311 [column 7, lines 23-26]. The request of URL is examined by the cache control unit 311 and the previously cached responses are analyzed to determine whether any of the cached responses are candidates for serving to the request [column 12, lines 57-61]. A matching URL cached entry is analyzed by the cached-response analyzer 306. Specifically the caching strategy flags, which were stored along with the cached response, are analyzed for applicability and for validity via the validity analyzer 315 [column 12, line 65 to column 13, line 3]. The validity analyzer 315 understands the cached response retrieval process and automatically invalidates pages based upon whether the candidate cached response is stale [column 13, lines 4-7].

The Examiner's citations are incomplete. In particular, Chamberlain insists that a cached response is only considered stale "after it becomes known that one of the source parts has been modified at the source," which is usually determined by "compar[ing] the candidate cached response's last modified date against all of the source parts' last modified dates" (column 12, lines 39-46; emphasis added). Thus, Chamberlain makes it clear that a cache system cannot invalidate cached data on its own. It must consider the 'last modified date' maintained by the source of the data.

In contrast, embodiments of Applicants' invention (claim 1, claim 6, claim 13) require no communication with a data source (e.g., a data server).

C. Chamberlain Requires an Explicit Invalidation Message from a Server to a Cache

The Examiner recognized that Chamberlain does not automatically invalidate a set of data stored in a cache system without awaiting an invalidation message from a data server (page 9, 3rd paragraph). However, the Examiner quoted and cited the same portions of Chamberlain as delineated above in Section I.B.

Contrary to the Examiner's assertion that "This information shows that the system automatically invalidates pages at the cache server in response to user request without awaiting an invalidation communication from the other data server," the cited portions of Chamberlain say nothing about an invalidation communication.

This is natural, because Chamberlain has already insisted that a cached response is only considered stale "after it becomes known that one of the source parts has been modified at the source," which is usually determined by "compar[ing] the candidate cached response's last modified date against all of the source parts' last modified dates" (column 12, lines 39-46; emphasis added). Thus, Chamberlain makes it clear that a cache system cannot invalidate cached data on its own. It must consider the 'last modified date' maintained by the source of the data, which requires communication with the source (i.e., an invalidation message).

II. Selected Claims

A. Claims 1-5

Claim 1 is directed to the automatic invalidation of data in a cache that is separate from a data server or web server that originates the cached data. As described above in section I.A, Chamberlain teaches away from this arrangement.

This aspect of the present invention was recited in claim 1 prior to this reply, and was therefore already searched, but no pertinent prior art was cited. And, no response to Applicants' argument in favor of this distinction was provided in the final office action. It is therefore assumed that no pertinent prior art teaches or suggests this feature. Allowance of claims 1-5 is therefore requested.

In addition, and as described in Section I.B, Chamberlain requires cooperation between a cache server and the source of data cached in the cache server. In contrast, claim 1 recites the automatic invalidation of cached data without communicating with the data server.

B. Claims 6-12, 18

Claims 6 and 18 were amended to include subject matter that was already recited in claim 1 prior to this reply. As described in Section I.A, Chamberlain does not teach or suggest automatic invalidation performed entirely in a cache system separate from a data server that provides data for caching in the cache system.

Further, as described in Sections I.B and I.C, Chamberlain requires some cooperation and communication between a cache server and a data source before the cache server can invalidate cached data. Claims 6 and 18 specify that the claimed method is performed at the cache system, and does not require a communication from the data server.

C. Claims 13-17, 19

Claims 13 and 19 recite the automatic invalidation of cached data, in a cache system, in response to a receipt at the cache system of a *change request* for the data, without waiting for a server to apply the change. This is very different from the claimed embodiment of Applicants' invention, in which a request to simply *view* a cached set of data is received. As described in Sections I.B and I.C, Chamberlain cannot invalidate a set of data at a cache server without interacting with a data server.

Claim 14 was rejected based on a portion of Ekanadham that addresses invalidation by one processor in a system in which processors share data that can be locked (column 2, lines 25-36). The system of Ekanadham is thus very different from the caching described in either Chamberlain or the present application. There are no memory locks placed on shared cached memory by different processors in claim 14, and so the "invalidation" mentioned in Ekanadham has no bearing on the invalidation of cached data being served in response to user requests.

D. Claims 20-25

Claim 20 describes a cache system that is coupled to a data server via a network link. In this embodiment of Applicants' invention, the data server and cache system are physically separate. As described in Section I.A, Chamberlain requires a cache and its data source to be colocated. Similarly, Ekanadham requires its cache to be colocated with a processor (e.g., FIG. 1). Thus, the cited prior art does not appear to be able to function with a cache system that is

separate from the source of the data, let alone teach or suggest the automatic invalidation of data cached in such a cache system.

The portion of Chamberlain cited by the Examiner (column 1, line 30 to column 2, line 5), describes a local area network depicted in FIG. 1. Mention is made of a volatile cache within a user's local computer system (column 1, lines 62-65). A local cache is unlikely to be considered a "cache system" by one skilled in the art, and cannot support the caching scheme of Chamberlain (which requires a cache closely coupled to its data source), let alone the automatic invalidation taught in Applicants' present invention.

CONCLUSION

No new matter has been added with the preceding amendments. It is submitted that the application is in suitable condition for allowance. Such action is respectfully requested. If prosecution of this application may be facilitated through a telephone interview, the Examiner is invited to contact Applicant's attorney identified below.

Respectfully submitted,

Date: May 20, 2004 By:  42,199

Daniel E. Vaughan (Registration No.)

Park, Vaughan & Fleming LLP
702 Marshall Street, Suite 310
Redwood City, CA 94063
(650) 474-1973: voice
(650) 474-1976: facsimile